ATIS-0800001

Verizon
40 Sylvan Road
Waltham, MA 02451-1128

Phone 781 466-0000
Fax 781 466-0000
muxiang.zhang@verizon.com

October 9, 2006

To: ATIS IPTV Interoperability Forum, DRM Task Group

Copy to: Tom Goode, ATIS Associate General Counsel

From: Muxiang Zhang
Verizon
40 Sylvan Road
Waltham, Massachusetts 02451
T: +1 781-466-3081
muxiang.zhang@verizon.com

On behalf of Verizon, I recently submitted a proposal to the ATIS IPTV Interoperability Forum, DRM Task Group. A copy of this proposal is attached for reference. In accordance with the ATIS Operating Procedure, section 10.4, the users attention is called to:

10.4.2.3 Notice

NOTE – The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights.
By publication of this standard, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. The patent holder has, however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the standards developer.

Sincerely

Muxiang Zhang

IIF CONTRIBUTION: IIF-DRM-2006-xxx

Date: October 3, 2006
Source: Verizon Communications, Inc.
Name: Muxiang Zhang
E-mail: muxiang.zhang@verizon.com
Phone: +1 781 466 3081
Title: IPTV Common Scrambling Algorithm Update

Intellectual Property Notice

NOTICE: *As an American National Standards Institute (ANSI) accredited standard development organization ATIS follows the patent policy developed by ANSI. ATIS strongly encourages the early disclosure of a patent that might be referenced in an ATIS standard or other deliverable. The party making the disclosure should identify:*

- *The patented invention; and*
- *An explanation regarding the relevancy of the patented invention to the work under development.*

If use of a patented invention is required for compliance with the standard, then ATIS would need to receive from the identified party or patent holder either assurance that:

- *A license will be made available without compensation to the applicants desiring to utilize the license for the purpose of implementing the standard; or*
- *A license will be made available to applicants under reasonable terms and conditions that are demonstrably free of any unfair discrimination.*

Please note that ATIS Operating Procedures prohibit the discussion of specific licensing terms within ATIS forums and committees. *In addition, neither ATIS nor the IIF shall be responsible for interpreting or making any determination concerning the validity, enforceability or scope of any patented invention referenced in or that may be relevant to any standard, guideline or other ATIS deliverable*

You may voluntarily provide the following information at the time that the Contribution is made or as soon as practicable:

Intellectual Property:
☒ Will License on Reasonable And Non Discriminatory (RAND) terms
☐ Will Not License
☐ Not Applicable (No IPR Declared)

Relevance of the IPR:
All or part of the contribution may be included in a patent application.

IIF CONTRIBUTION: IIF-DRM-2006-xxx

Date: **October 3, 2006**

Source: **Muxiang Zhang, Verizon Communications, Inc.**
Muxiang.zhang@verizon.com, +1 781 466 3081

Title: **IPTV Common Scrambling Algorithm Update**

1 Introduction

This document updates the proposed IPTV Common Scrambling Algorithm as described in IIF-DRM-2006-430. The goal of the update is to make the proposed IPTV Common Scrambling Algorithm purely based on AES CBC mode as specified in FIPS PUB 81 and NIST Special Publication 800-38A, while keeping the updated IPTV Common Scrambling Algorithm as much compatible as possible with scrambling algorithms specified by other standards bodies, e.g., ANSI/SCTE-52 and ATSC. As in IIF-DRM-2006-430, the updated IPTV Common Scrambling Algorithm uses AES under Cipher Block Chaining (CBC) mode to scramble MPEG-2 transport stream packets. Only the payload of MPEG-2 transport stream packet is scrambled, the header and the optional adaptation field are not scrambled.

Note that the updated IPTV Common Scrambling Algorithm can also be used to scramble MPEG-4 transport stream packets.

2 Scrambling of MPEG-2 Transport Stream Packet

Fig. 1 illustrates the scrambling of a MPEG-2 transport stream packet. In Fig. 1, E_k denotes the AES encryption algorithm under the control of a scrambling key, K, IV denotes the initialization vector, and $\oplus$ denotes bit-wise exclusive-OR operation. Both the scrambling key K and the Initialization Vector IV consist of 16 bytes or 128 bits.

To scramble a MPEG-2 transport stream packet, the payload is divided into blocks of 16 bytes. The last block may consist of less than 16 bytes. As shown in Fig. 1, let $P_1, P_2, \ldots, P_m$, $m \geq 1$, denote the plaintext blocks of the payload of a MPEG-2 transpost stream packet, where P_1 is the moste significant block and P_m is the least significant block. Correspondingly, let $C_1, C_2, \ldots, C_m$ denote the ciphertext blocks. Also, let τ denote the lenth of of P_m, i.e., the number of bits that P_m consists of. Base on the value of τ, the scrambling procedure can be described mathematically as follows.

If τ is equal to 128 bits, then

$$C_i = E_K (C_{i-1} \oplus P_i), \text{ for } 1 \leq i \leq m, C_0 = IV.$$

If τ is less than 128 bits and m is greater than 1, then

$$C_i = E_K (C_{i-1} \oplus P_i), \text{ for } 1 \leq i < m, C_0 = IV,$$

and

$$C_m = [E_K (C_{m-1} \oplus H)]_\tau \oplus P_m,$$

Where $[x]_\tau$ denotes the least significant τ bits of x, and H is a constant equal to the following hexadecimal number:

$$H = 0x7884fe536c3588b73c2f604e4813fbe1$$

If τ is less than 128 bits and m is equal to 1, then $C_1 = [E_K (IV \oplus H)]_\tau \oplus P_1$. This describes the scrambling of a solitary termination block, which will be discussed further Section 5.

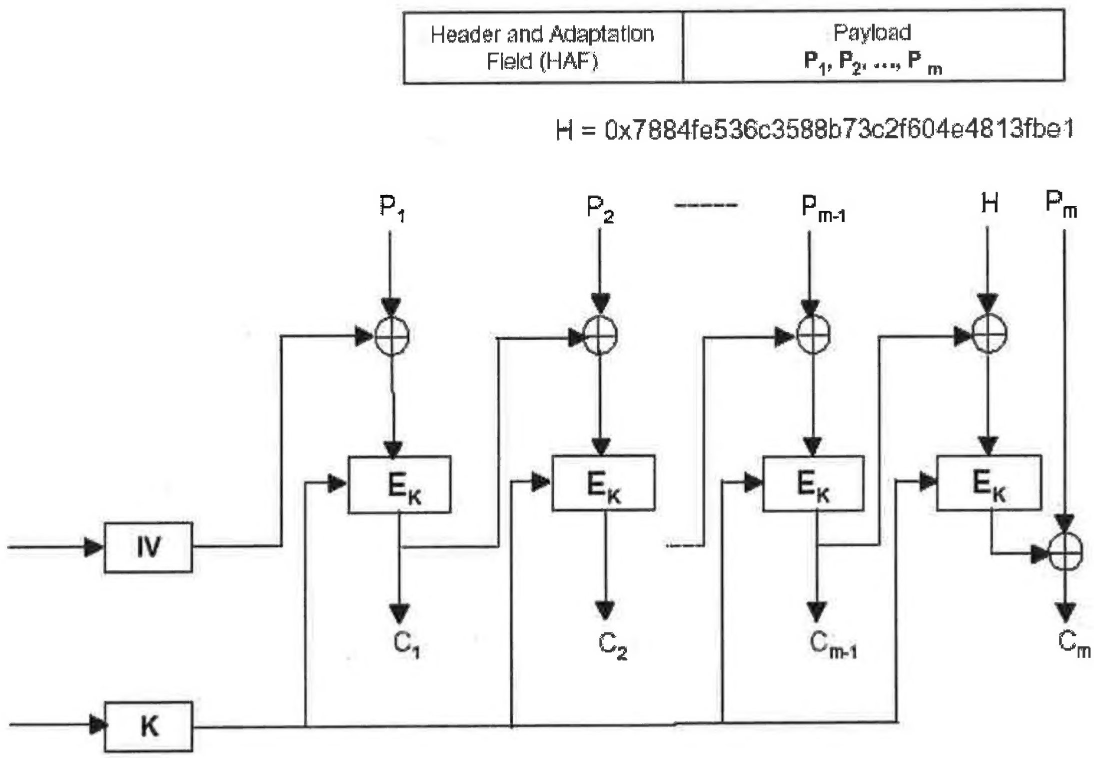

Fig. 1. Scrambling of MPEG-2 Transport Stream Packet

3 Generation of Initialization Vector (IV)

Fig. 2 illustrates the generation of the Initialization Vector, IV. Note that IV can be set either to a constant or to a programmable random number. In Fig. 2, f_{IV} is a one-bit flag controlling the generation of IV. When f_{IV} is equal to zero, IV is set to be all zero. When f_{IV} is equal to 1, the AES encryption algorithm E_k first encrypts zero under the control of the scrambling key K. Then, the AES encryption algorithm encrypts zero again under the control of a new key, $E_k(0)$, and the resultant ciphertext is assigned to IV. By default, f_{IV} is set to 1. Optionally, a network operator may chose to use its own mechanism to set f_{IV} to different values, e.g., setting f_{IV} to 0 during the bootstrapping of receiving devices.

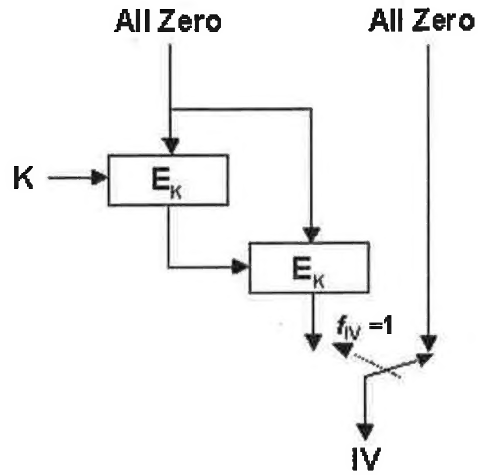

Fig. 2. Generation of Initialization Vector (IV)

4 Descrambling of MPEG-2 Transport Stream Packet

Fig. 3 illustrates the descrambling procedure of an MPEG-2 transport stream packet. In Fig. 3, D_k denotes the AES decryption algorithm under the control of a descrambling key, K, which is equal to the scrambling key, and IV denotes the initialization vector. Both the descrambling key, K, and the Initialization Vector, IV, consist of 16 bytes or128 bits.

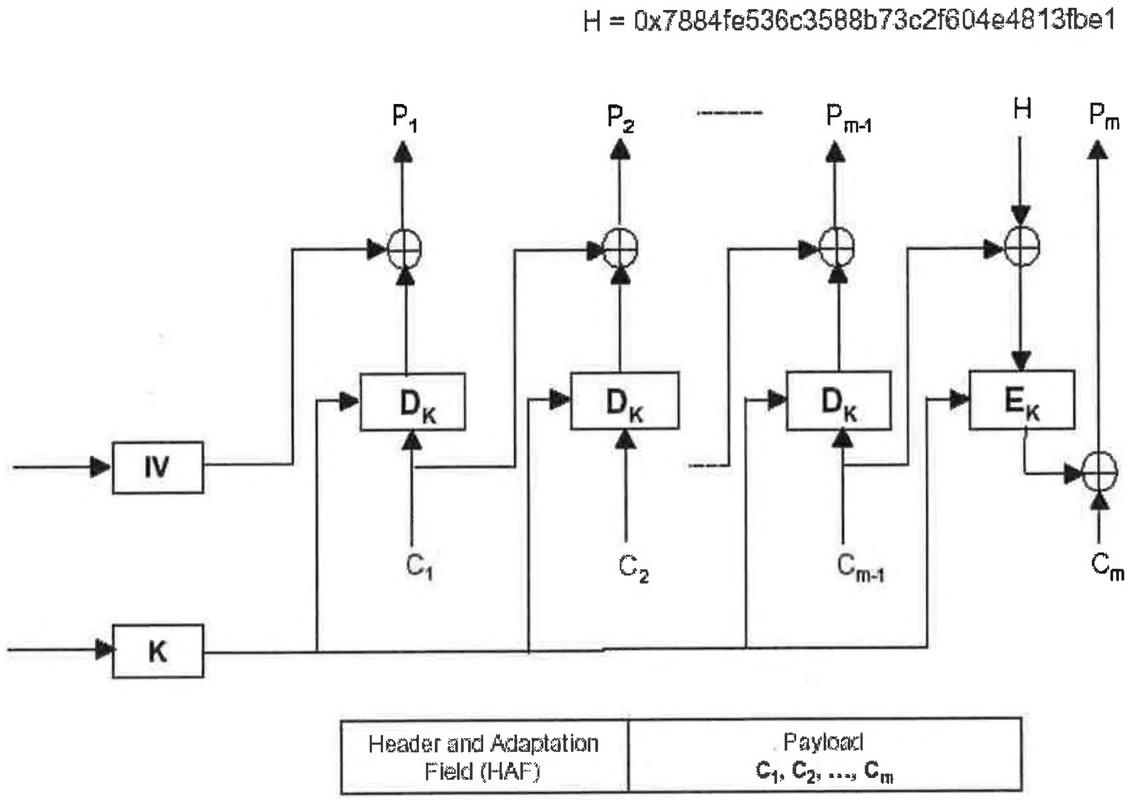

Fig. 3. Descrambling of MPEG-2 Transport Stream Packet

To descramble a MPEG-2 transport stream packet, the payload of the MPEG-2 transport stream packet is divided into blocks of 16 bytes. The last block may consist of less than 16 bytes. As shown in Fig. 3, let $C_1, C_2, \ldots, C_m$, $m \geq 1$, denote the ciphertext blocks and let $P_1, P_2, \ldots, P_m$, denote the corresponding plaintext blocks. Also, let τ denote the number of bits that C_m consists of. Base on the value of τ, the descrambling procedure can be described mathematically as follows.

If τ is equal to 128 bits, then

$$P_i = D_K(C_i) \oplus C_{i-1}, \text{ for } 1 \leq i \leq m, C_0 = IV.$$

If τ is less than 128 bits and m is greater than 1, then

$$P_i = D_K(C_i) \oplus C_{i-1}, \text{ for } 1 \leq i < m, C_0 = IV,$$

and

$$P_m = [E_K(C_{m-1} \oplus H)]_\tau \oplus C_m,$$

Where $[x]_\tau$ denotes the least significant τ bits of x, and H is a constant equal to the following hexadecimal number:

H= 0x7884fe536c3588b73c2f604e4813fbe1

If τ is less than 16 bytes (or 128 bits) and m is equal to 1, then $P_1 = [E_K(IV \oplus H)]_\tau \oplus C_1$. This describes the descrambling of a solitary termination block, which will be discussed further in Section 5.

5 Processing of Solitary Termination Block

Due to the varying length of the optional adaptation field, an MPEG-2 transport stream packet may contain a very small payload of less than 16 bytes. In such a scenario, the payload would be the first and last the block, which is called a solitary termination block. Fig. 4 describes the scrambling of a solitary termination block, denoted by P_1. Let τ denote the number of bits that P_1 consists of. Then the ciphertext block C_1 is computed by taking the bit-wise exclusive-OR of P_1 and the least significant τ bits of $E_K(IV\oplus H)$.

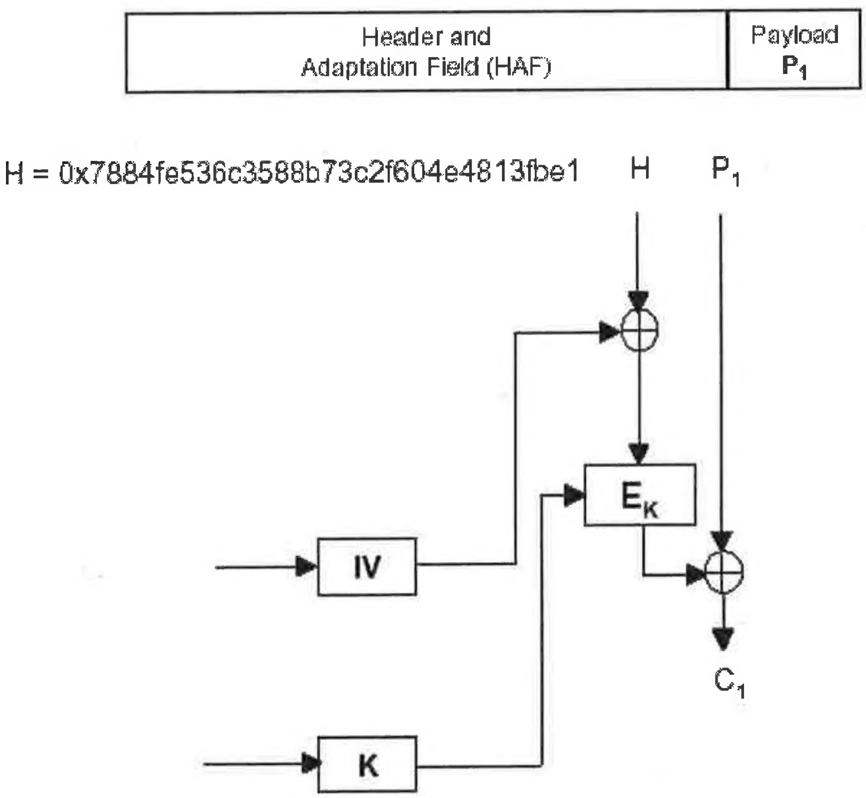

Fig. 4. Scrambling of Solitary Termination Block

The descrambling of a solitary termination block is the same as the scrambling. Fig. 5 describes the descrambling of a solitary termination block, denoted by C_1. Let τ denote the number of bites that C_1 consists of. The plaintext block P_1 is computed by taking the bit-wise exclusive-OR of C_1 and the least significant τ bits of $E_K(IV\oplus H)$.

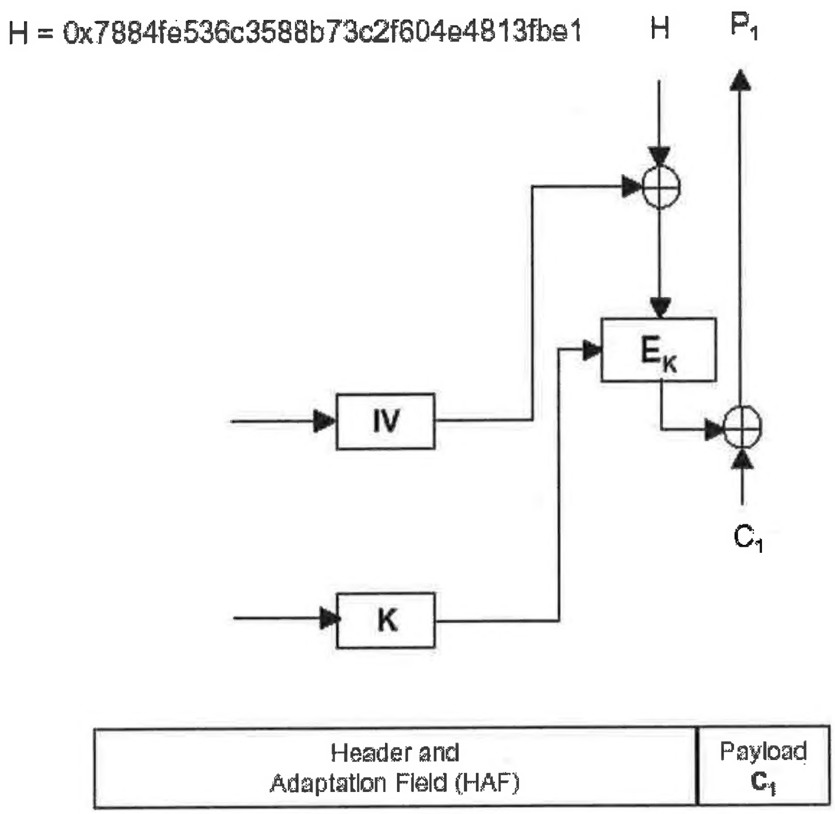

Fig. 5. Descrambling of Solitary Termination Block